

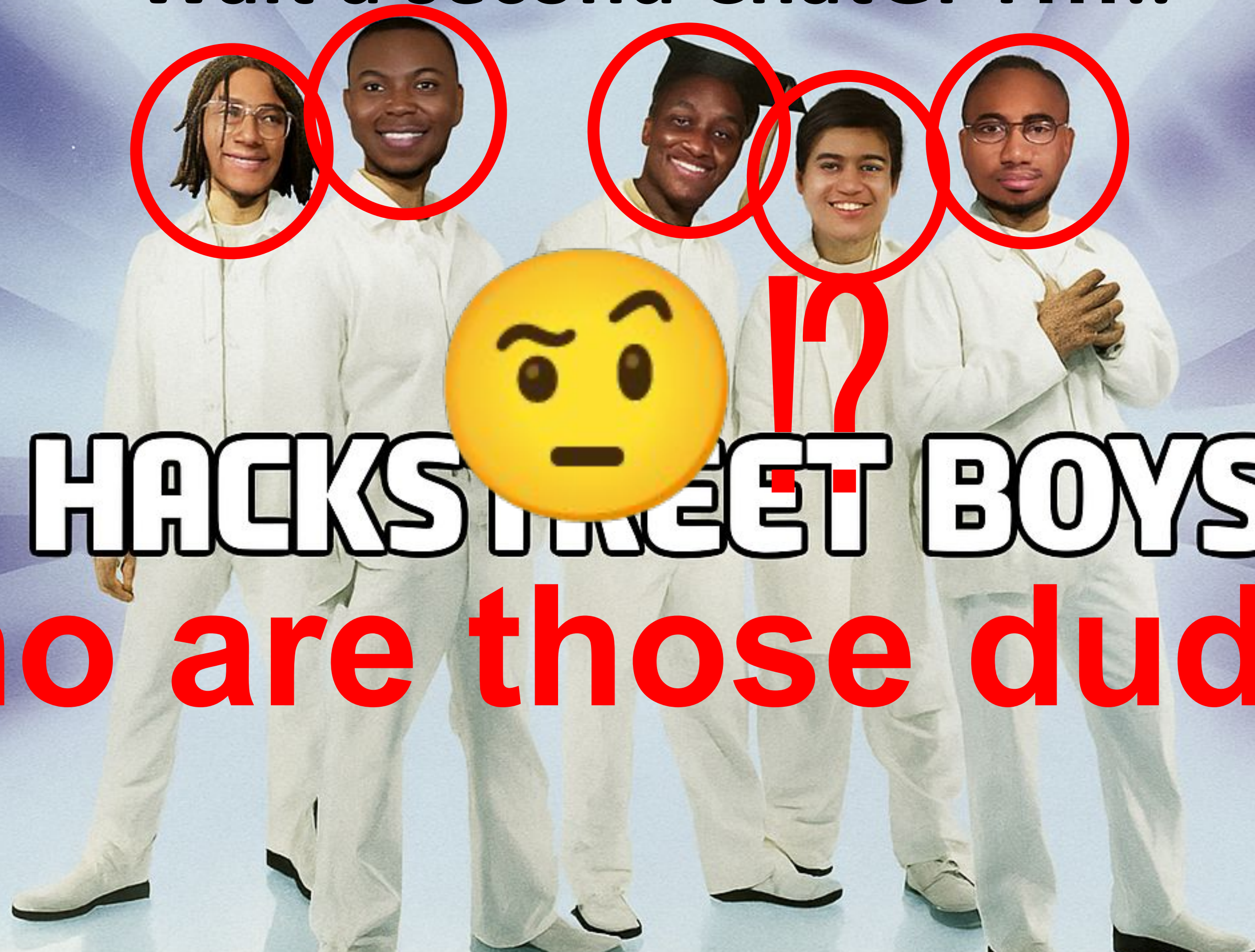
HACKHPC@
ADMI25
HACKATHON

hackhpc.github.io/admi25

HackStreet Boys



Wait a second ChatGPT.....



HACKS IN THE STREET BOYS

Who are those dudes?



YARI PETTIS

ZION PEASE

DAVE BROWN

EJAY AGUIRRE

JULIAN TOLBERT



Project Overview & Team Roles

Julian – GitHub & Documentation Lead

Sets up repo, manages README.md, folder structure, and code organization.

Ejay – Poster & Presentation Lead

Designs project poster and final slide deck; supports portal content & layout. Will be working on Flask as well.

Zion – Paper Analyst

Selects target papers, evaluates reproducibility criteria, leads scorecard writing.

Dave – Code Runner

Attempts to reproduce paper results, logs code, dataset, and hardware issues

Yari – Communications & Submission Manager

Coordinates daily check-ins, manages submission proof, team info, and final review.



5 Day Plan

Days

Focus

Key Outputs

Monday

Kickoff & Setup

Repo setup, roles assigned, paper shortlist

×

×

×

×

Tuesday

Paper Deep Dive & Planning

Paper selected, access tested, plan slides

Wednesday

Scorecard Development & Testing

JSON/CSV file, initial portal layout, graph

Thursday

Portal Build & Poster Finalization

Site live, poster PDF, submission proof

Friday

Final Presentation & Deliverables
Wrap-up

Slides PDF, final push to GitHub, rehearsal

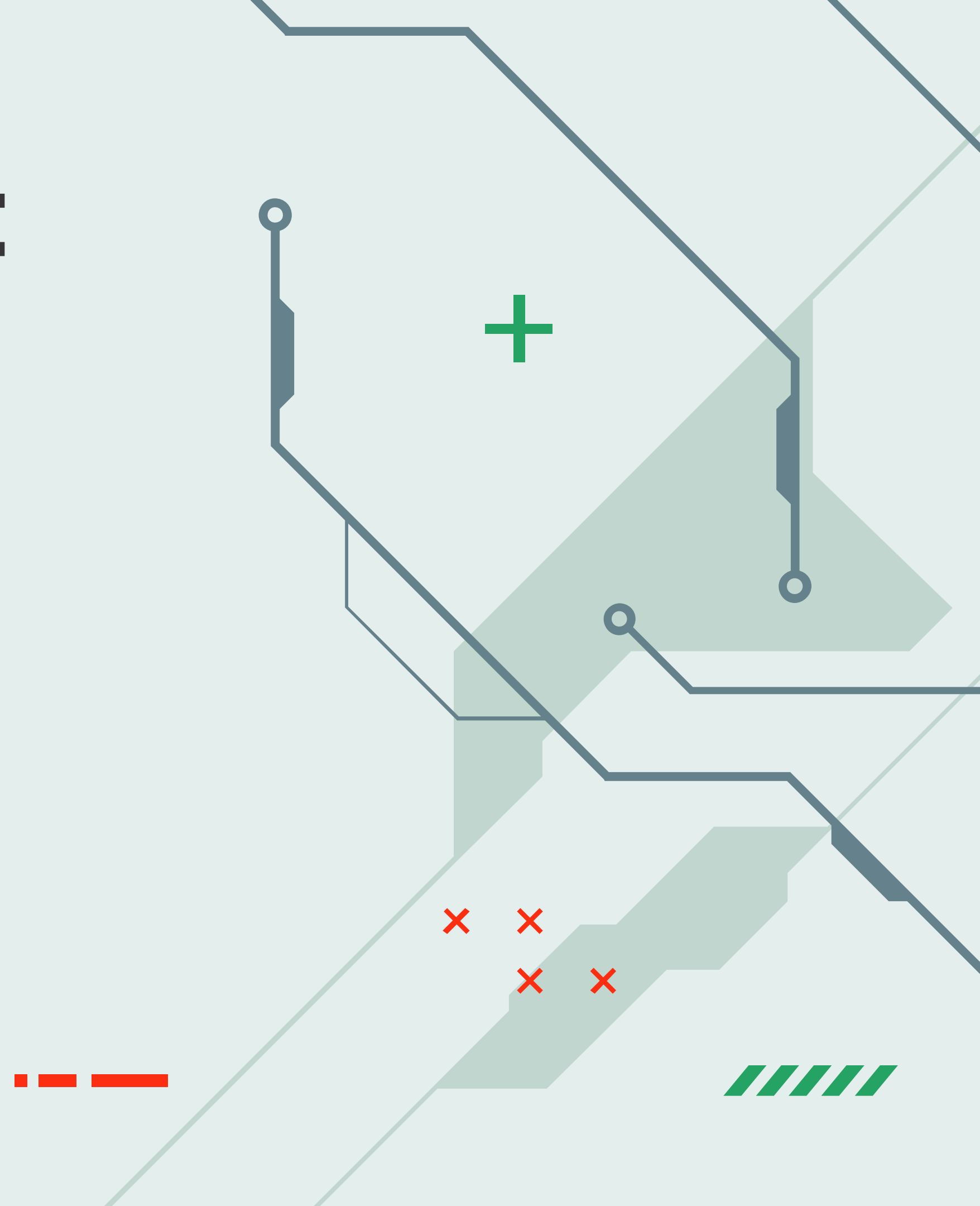
HackStreet Boys: Project Progress

Priorities:

Our current priority is creating the datasets from the papers and a proper rubric.

We have decided to use a python script to read the scores in so that we can use the data to plot that information.

While also starting to build our website/github to host our information and the actual score cards that we are building.



HackStreet Boys

Project Plan

Update!

The project plan before is still in progress but we have started to move somethings to be done in tandem with other goals, like the beginning to code our flask and properly setting up the information held in papers/websites. While also including more flexibility due to time zones and some skill levels.

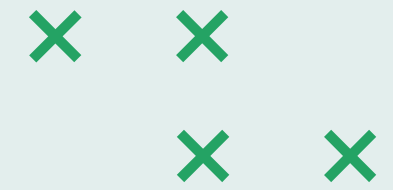
— — .

Technology/Problems

We have been using AI like manus, chatgpt and gemini prompting it to give us ideas for rubric and helping debug our code. While also applying our skills in virtual environments like colab to test out our python code.

We created a python script that scrapes a specified number papers, feeds them into Gemini AI, which then provides feedback on some scoring metrics and stores them in a csv file.





HackStreet Boys

We were able to create a python script to evaluate papers and put them into a dataframe to be graphed. We also have started to get our flask app up and running including our team/background information

```
# 2. Iterate through each paper and get LLM evaluation
for i, paper in enumerate(papers_for_evaluation):
    paper_title = paper.get("Title", "Untitled Paper").replace('\n', ' ')
    print(f"\n--- Evaluating Paper {i+1}: '{paper_title}' ---")

    prompt = generate_llm_prompt_for_paper(paper, RUBRIC)

    try:
        response = llm_model.generate_content(prompt)
        evaluation_result_text = response.text.strip()

        print(evaluation_result_text)
        all_raw_evaluations_text += f"\n--- Evaluation for Paper {i+1}: '{paper_title}' ---\n"
        all_raw_evaluations_text += evaluation_result_text
        all_raw_evaluations_text += "\n" + "="*80 + "\n"

        # Parse the LLM's text response directly into a dictionary
        parsed_scores = parse_llm_response_to_dict(evaluation_result_text, paper_title)
        structured_evaluations_list.append(parsed_scores)

    except Exception as e:
        print(f"Error evaluating paper '{paper_title}' with Gemini: {e}")
        error_message = f"LLM evaluation failed for this paper: {e}"
        all_raw_evaluations_text += f"\n--- Evaluation for Paper {i+1}: '{paper_title}' (FAILED) ---\n"
        all_raw_evaluations_text += error_message
        all_raw_evaluations_text += "\n" + "="*80 + "\n"
        structured_evaluations_list.append({"Paper Title": paper_title, "Evaluation Status": "Failed"

```


HackStreet Boys

Rubric:

This rubric assesses scientific presentations based on **six key criteria**, each scored from 1 (Poor) to 4 (Excellent):

1. Accessibility

Evaluates whether the resource is open and free or requires payment/login.
2. Computer Requirements

Measures how reasonable and clearly specified the hardware/software needs are.
3. Reproducibility of Results

Assesses how easily others can reproduce the project using provided steps and data.
4. README File

Looks at the presence and quality of documentation explaining the project’s purpose, setup, and usage.
5. GitHub Repository

Checks for an organized, public repo with code and version control.
6. Coding Software Availability

Rates how accessible and well-documented the required coding tools are.

- Total Score: /24
- Excellent: 21–24

• Good: 17–20

• Fair: 13–16

• Needs Improvement: Below 13



EVALUATION RUBRIC

	CRITERIA	EXCELLENT (4 pts)	GOOD (3 pts)	FAIR (2 pts)	POOR (1 pt)
1. ACCESSIBILITY	Fully open acces, no pagment or	Mostly specified access or mini- mal barrier	Mostly specified access or milltl mal barrier	Reqlures paid access or difficult repr.	Not accessible
2. COMPUTER REQUIREMIENTS	Clearly specified and reasonable for most users	Some-specificat- lons provided (e.g.)	Some partially specified	Steps unclcar or data missing	Not specified or linrealistic
3. REPRODUCIBILITY OF RESULTS	All steps and data clearly documented.	Steps unclcar or data missing	Steps unclcar or data missing	Not reproducible, Steps unclcar	No README repo
4. README FILE	Clear detailed README explaining purpose setup. Ubsibh, to disperitetviss	README missing, unreadable. in- complete content	No README missing, un- complete content	No Github misnno	No Github repo
5. GITHUB REPOSITORY	Public aithub repa with arg- anized code. and with documented	*Software/tools used are free. opem- source, and	*Software/tools used are free or undocumented	Coding missno available, und- ocumcvented	Coding tools not available or undocumen
6. CODING SOFTHARE AVAILABILITY	TOTAL POINTS: ----- / 24 Optional Scoring Guide: Excellent: 21-24 Good: 17-20 Fair: 13 Needs Improvement below 13				



EVALUATION RUBRIC

	CRITERIA	EXCELLENT (4 pts)	GOOD (3 pts)	FAIR (2 pts)	POOR (1 pt)
1. ACCESSIBILITY	Fully open acces, no payment or	Mostly specified access or mini- mal barrier	Mostly specified access of mütim mal barrier	Reouires paid access or difficult repr:	Not accessible
2. COMPUTER REQUIREMENTS	Clearly specified and reasonable for most users	Some specificati- ons provided (e.g.)	Some partially specified	Only portially specified	Not specified or unrealistic
3. REPRODUCIBILITY OF RESULTS	All steps and data clearly documented	Steps unclcar or data missing	Steps unclcar or data missing	Not reproducible; steps unclcar	No RAIDME repo
4. README FILE	Clear detailed README explaining purpose setup. U/sable no dependencies,	README missing unreadable, inco- mplete content	No README missing. un- mpetable content	Coding nissno provided	No Github repo
5. GITHUB REPOSITORY	Public github repo with organized code, version ard with documentation	Software/tools used are free, open- source, and	-Software/touls used are free or undocumented	Coding tools no avoailable, unclou or undvcumented	Cooing tools not araj lab e urdoutmen.
6. CODING SOFIWARE AVAILABILITY	TOTAL POINTS: _____ / 24 Optional Scoring Guide: Excellent: 21-24 Good: 17-20 Fair: 13- Needs Improvement below 13				

Website Demo

The background features abstract geometric shapes in light green and dark blue-grey. In the top-left, a dark blue line with a small circle at its end extends from the edge. In the top-right, a similar line with a circle at its end extends from the edge. In the bottom-left, a dark blue line with a circle at its end extends from the edge, and a small red dashed line is positioned below it. In the bottom-right, a cluster of four green 'x' marks is visible.